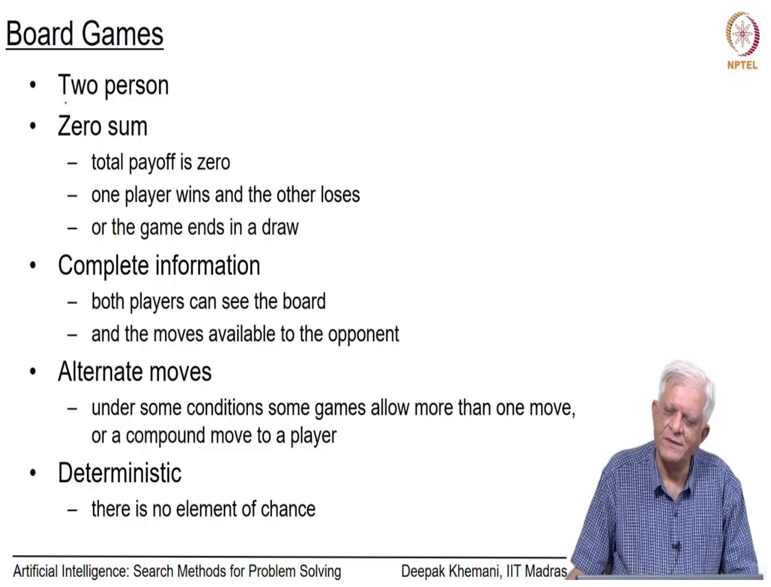
Artificial Intelligence: Search Methods for Problem Solving
Prof. Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Chapter – 08
A First Course in Artificial Intelligence
Lecture – 52
Game Playing
Board Games and Game Trees

So, so far we have looked at this field of study called Game Theory which looks at rational behaviour in multi-agency scenarios. Then we looked at a host of popular games and got some history about which games are played well by machines and which are not played well by machines.

In the subsequent lecture, we will focus on simpler games. These are called board games and we will see how they are modelled as game trees and what are the algorithms used for playing these games.

(Refer Slide Time: 00:57)



Board Games

- Two person
- Zero sum
  - total payoff is zero
  - one player wins and the other loses
  - or the game ends in a draw
- Complete information
  - both players can see the board
  - and the moves available to the opponent
- Alternate moves
  - under some conditions some games allow more than one move, or a compound move to a player
- Deterministic
  - there is no element of chance

Artificial Intelligence: Search Methods for Problem Solving Deepak Khemani, IIT Madras

So, board games are essentially two person games; in the sense that they are played between two players. So, chess is an example Checkers, Go or the humble cross and noughts which children play – all these are examples of board games in which there are two people involved.

These are zero sum games where one side wins and the other side loses. The total payoff as we have studied in zero sum games is zero or the game can end in a draw which is and they get zero points for that and as we will see shortly.

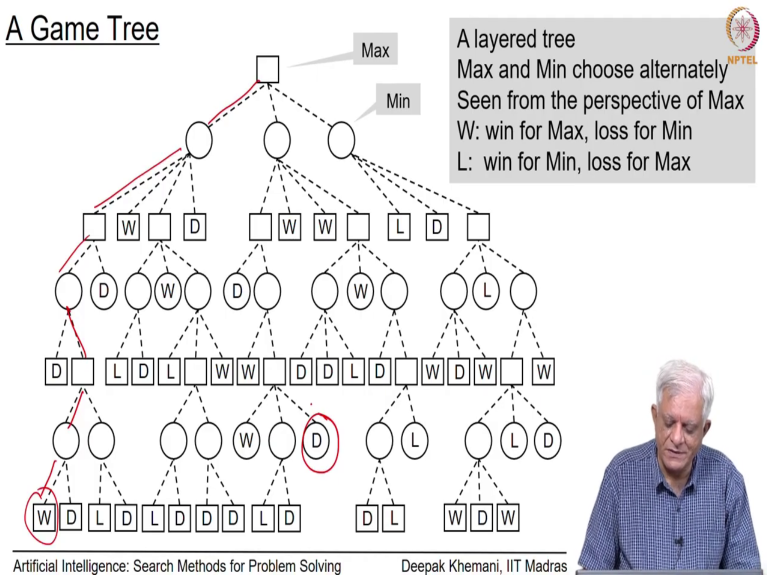
These are complete information games where both players can see the board position completely and they can see the moves available to the opponent as well. So, they know exactly what the other person can do and take that into account in their own strategy formation. These are typically alternate move games where each player makes a move turn by turn. Though under certain conditions one player maybe allowed more than one move or what can we call as a compound move.

So, if you played the game of checkers you know that you can capture two coins at one time or if you have looked at Reversi or Othello also as it is called, then if the opponent does not have a move you get a second move. But, typically they are alternate move games that you that you play a move, I play a move and so on and they are deterministic games there is no element of chance of any kind involved in this.

So, as you can see that they are the simplest kind of games that we can consider, but in this small set our games like Chess and Go which have fascinated us for centuries together.

It does not include games like backgammon which has a throw of throw of dice or card games like poker and bridge or in games like scrabble, where there is an element of chance as to you know which letters you get and you do not know what the opponent is opening. These are much simpler games – two persons, zero sum, complete information, alternate move, deterministic games. So, let us see how to tackle such games.

(Refer Slide Time: 03:12)



A Game Tree

A layered tree
Max and Min choose alternately
Seen from the perspective of Max
W: win for Max, loss for Min
L: win for Min, loss for Max

Max
Min
W
D
L
Artificial Intelligence: Search Methods for Problem Solving
Deepak Khemani, IIT Madras
NPTEL

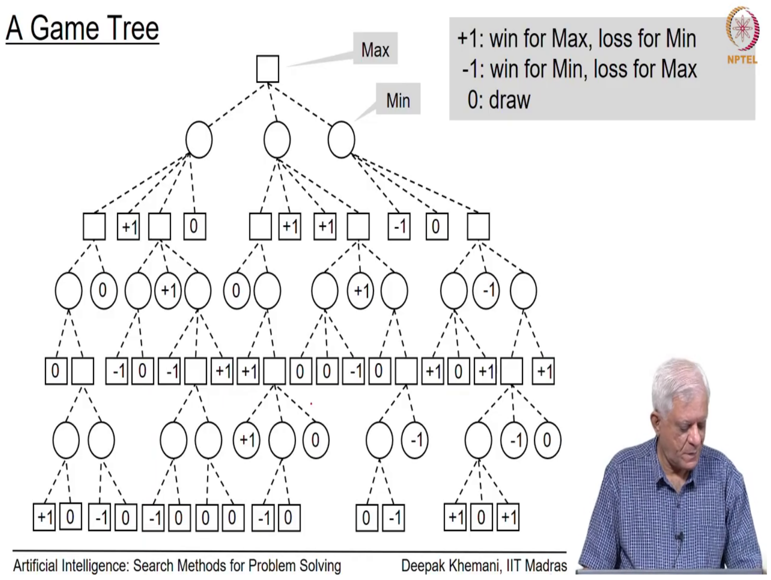
Once you know the rules of a game, then you can construct a corresponding tree which is called the game tree. In this tree there are two players involved. Remember it is a two person game and the two players are signified by different ways of drawing the node and it is a layer tree in which we always look at the game from Max's perspective. The one player is called Max the other player is called Min typically we draw the Max as a square and Min as a circle.

And, it is a layer tree in the sense that at the first level Max has to make a choice. Then based on the choices of Max at the second layer Min makes a choice and based on the choice of Min at the third layer Max again makes a choice and so on and so forth, and all the games that are possible are represented as paths in this game tree. So, Max and Min choose alternately; Max makes the first move, Min the second and so on.

We always see the game tree from the perspective of Max because we will imagine that we are going to write a program for Max. The leaves of the game trees are labelled with the outcome of the tree. So, once a game is ended you know what has happened uh. So, W stands as a win for Max and loss for Min. So, for example, there is a leaf which is labelled W which means that if the game that is played is as follows.

Then Max would have won the game the labelled W stands for win and the win is for Max; L stands for a win for Min and the loss for Max. So, if we had ended up in a leaf node which is labelled L that would mean that Min has won the game and we have the third outcome possible which is for example, here which is draw D stands for a draw which means neither side is one essentially.

(Refer Slide Time: 05:38)



A Game Tree

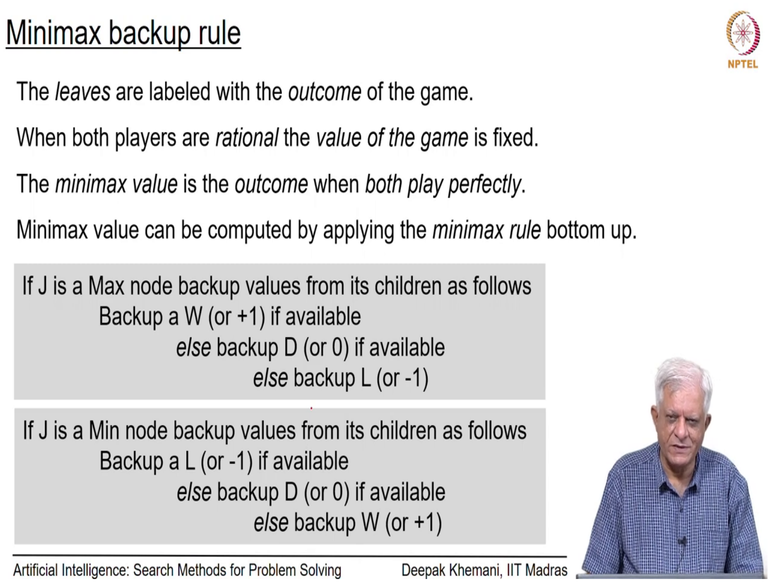
+1: win for Max, loss for Min
-1: win for Min, loss for Max
0: draw

Max
Min
NPTEL
Artificial Intelligence: Search Methods for Problem Solving
Deepak Khemani, IIT Madras

So, to kind of quantify the payoffs we can sometimes label them with plus 1 for Max minus 1 for Min and 0 for the draw. So, you can see that the total payoff is if one of the side wins it is plus 1 and minus 1 for each player you have to add them up and if either side wins then it is a draw. This the both the gets zero points essentially.

You can also see why Max is called max because Max is trying to reach the score of plus 1 which is a maximum of the three possible outcomes and Min is called Min because Min is trying to choose the smallest value in the game which is minus 1.

(Refer Slide Time: 06:20)



Minimax backup rule

The *leaves* are labeled with the *outcome* of the game.

When both players are *rational* the *value of the game* is fixed.

The *minimax value* is the *outcome* when *both play perfectly*.

Minimax value can be computed by applying the *minimax rule* bottom up.

```
If J is a Max node backup values from its children as follows
        Backup a W (or +1) if available
                else backup D (or 0) if available
                        else backup L (or -1)
```

```
If J is a Min node backup values from its children as follows
        Backup a L (or -1) if available
                else backup D (or 0) if available
                        else backup W (or +1)
```

Artificial Intelligence: Search Methods for Problem Solving — Deepak Khemani, IIT Madras

Now, what happens when a game like this is played? Now, remember that we have said that we are concerned with rationality here that we assume that both the players are perfectly rational, they are perfectly able to think about the consequences of their actions. They can look ahead and analyze the complete game tree.

So, given a board game like this we can determine what is the Max equilibrium for this particular game and which is when both the players play perfectly. So, as we said the leaves are labelled with the outcome of the game when both players are rational the value of the game becomes fixed; because both players make the best move at every stage. The that value is known as the minimax value it is a outcome and both the players play perfectly.

And, the minimax value of a game or a game tree can be computed by applying the minimax rule in a bottom up fashion which means you start from the leaves and you go up towards the root of the game tree. The minimax rule is as follows that if J the node that you are looking at is a Max node, then back up values from it is children as follows that if a W node is available you back it up or plus 1 as the case may be.

If W is not available you backup 0 or d else you backup L or minus 1; which means that if there is a move in which Max can win, Max will make that move. And, therefore, the value of that game will be W otherwise Max will make a move in which the game becomes a draw. So, the value of that node J will become 0 backed up value otherwise there is no option, but to lose the game.

Likewise for Min you back up in the reverse order that if you are a Min node if you are looking at if J is a Min node, then from the children of J you backup L if possible. Remember L is a loss for Max and a win for Min and Min is also trying to win the game.

So, L also corresponds to minus 1 that we said and that is why we said that Min player is called so because it chooses the minimum value which is minus 1 in this case. If it cannot get a value of minus 1 it will also like Max settle for a draw. So, the second choice is always 0 or draw for both the players and the third choice is to lose the game in which case it will have to backup a value of W which is a win for max, but a loss for this thing.

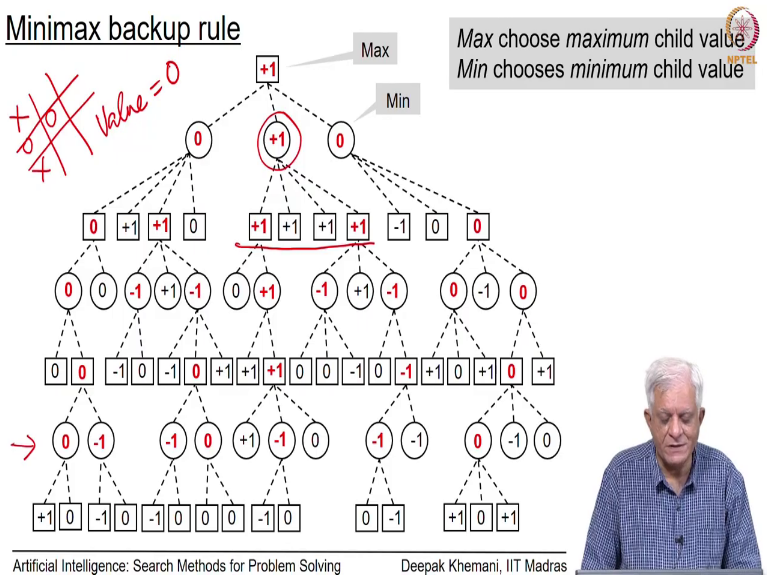
So, at each layer we can do this backup of values and once you do this, we will get the minimax value of the game tree. So, let us see this process on the small game tree that we have constructed.

Now, remember that the game tree encapsulates all the rules of the game you have constructed the game tree by looking at the game rules and deciding what are the moves for Max. And what are the moves for Min we have not labelled the moves in our game tree, but you can imagine that each move has a label.

For example, pawn looking 4 or place a cross at a particular square in cross and noughts whatever the case may be, but eventually we have a game tree where there are set of choices available, it is a layer tree. And, now we are trying to find out what is the minimax value of the game tree and we will do that by applying this backup rule in a bottom up fashion.

We will start with the leaves and move up towards the root of the game tree and having done that we will know that if both the players play perfectly what is the outcome of the game. So, let us look at this for the small game tree that we constructed.

(Refer Slide Time: 10:31)


Minimax backup rule
Max
Min
Max choose maximum child value
Min chooses minimum child value
Value = 0
Artificial Intelligence: Search Methods for Problem Solving
Deepak Khemani, IIT Madras

As we said we will start doing this process in the from a bottom of fashion. So, if you see this node at the lowest the lowest internal node that we can find. It has two choices – one is plus 1 and the other is 0. Now, clearly this is a Min node, it will choose the Min of the two values. So, it will choose a value of 0. Its sibling has two choices which is, minus 1 and 0. So, it will choose minus 1 and in this fashion, we will back up the values and eventually all the nodes will get a value.

So, Max chooses a maximum child value and Min chooses the minimum child value and this is the process that we are seeing now. First the nodes at the lowest layer the internal are being backed up and then once they have their values their values are backed up to the next higher layer and so on essentially. So, you can see that at each stage Min is choosing the minimum of all possible values and Max is choosing the maximum of all possible values.

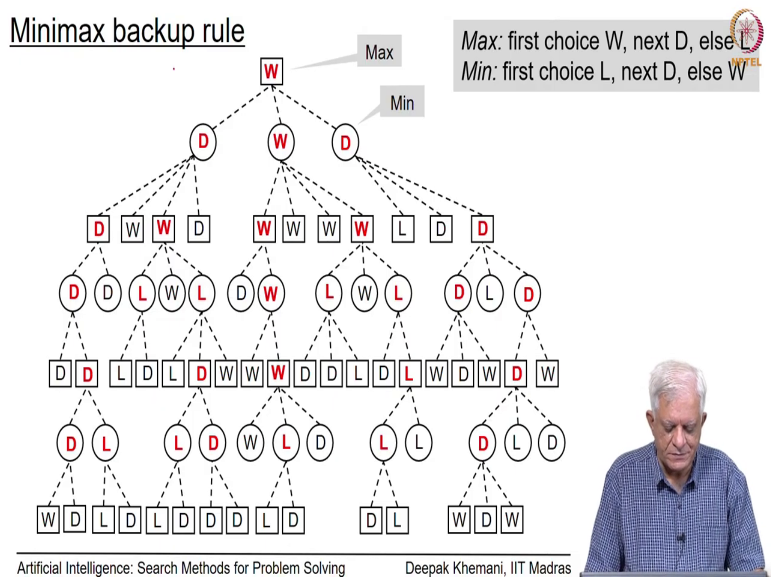
Sometimes as you can see Min is forced to choose plus 1; which is a win for Max and that happens for example, here when all the children of that particular Min node are labelled as plus 1. So, it has no choice, but to back up a plus 1. So, this small game tree that we constructed somewhat random tree is such that a both the players play perfectly then Max will win the game essentially.

Obviously, if a game tree was so small and you could analyze it completely, it would be no fun to play the game because you are assuming that both the players are perfect, but as we will see most games are not so small and so, they are not easy to analyze which is why games like Chess and Go still fascinate us.

But, a game like Tic-tac-toe or cross and noughts which you might remember is a game it is played on a 3 by 3 square. And essentially it involves placing a cross or placing a nought and then placing a cross and the nought and so on and so forth.

Now, those of you who have had the chance of playing this game in your childhood you would know that the value of this game is 0. Which means that if your opponent is playing well, if both the players are playing perfectly, then the game will always end in a draw; which is why, as you grow older you lose interest in the game. Because you know that there is nothing much which can be done, but that is not the case for chess as you will see.

(Refer Slide Time: 13:36)



Minimax backup rule

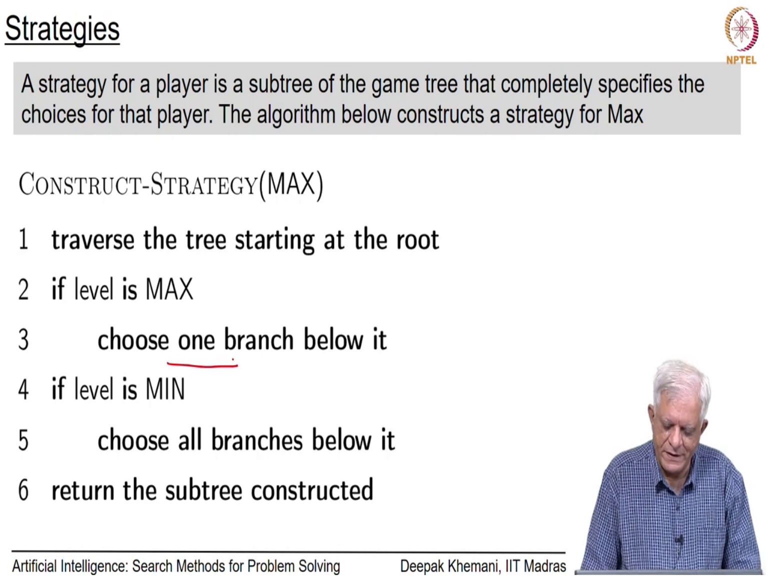
Max: first choice W, next D, else L
Min: first choice L, next D, else W

Artificial Intelligence: Search Methods for Problem Solving   Deepak Khemani, IIT Madras

You could have use the nomenclature of the tree labels win a draw or loss W, D or L. So, Max's first choice is W, if not W then D, else L and likewise Min's first choice is L, if not D, else W. So, this tree that we are looking at now it is just relabelled. You could have just substituted minus 1 with L, 0 with D and plus 1 with W and you would have got the same tree.

But, very often as human beings we like to look at the words like win or loss and because that is how we think of in terms of the game, but the two notations are equivalent. As I move forward we will see that we will divide we will have a tendency to move towards the numerical notation or reasons that we will see which will become clear as we go along.

(Refer Slide Time: 14:34)



Strategies

A strategy for a player is a subtree of the game tree that completely specifies the choices for that player. The algorithm below constructs a strategy for Max

```
CONSTRUCT-STRATEGY(MAX)
1  traverse the tree starting at the root
2  if level is MAX
3       choose one branch below it
4  if level is MIN
5       choose all branches below it
6  return the subtree constructed
```

Artificial Intelligence: Search Methods for Problem Solving   Deepak Khemani, IIT Madras

Now there is a notion called strategy. Remember, that when we talked about game theory we said what is the strategy for each player. It is the rational choice that you can make. Now, games that we talked of like business dilemma they were you had to make only one move. So, you have to simply have a device of whether you will betray your partner in crime or whether you would cooperate with that this thing. So, you have to make only one move.

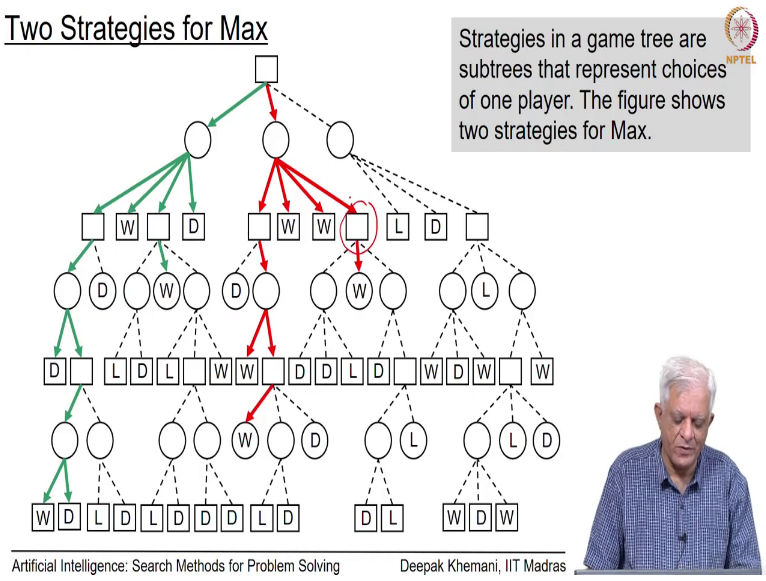
But, in multiplayer games you have to make a sequence of moves. You make you make you make in a multi move game you make a move then opponent makes a some move, then you make a move and then opponent makes and move and so on. So, in such a scenario what is the strategy? So, we define a strategy for the player as a sub tree of the game tree, such that it completely specifies the choices for that particular player.

So, once I have looked at a game and say in this board position this is a move I will make, then you are freezing your choices, but because you do not know what the opponent will do, you cannot decide for the opponent. So, you have to consider all possible choices for the opponent. So, this is how a strategy is constructed and we have again going to look at it from the perspective of Max; because we imagine we are writing a program which will start playing the game.

So here is a small program for constructing a strategy for Max. You start traversing the game tree starting at the root, go down level by level if it is a MAX node that you are looking at or if it is a MAX level, then choose one branch below it. If it is a MIN node you are looking at, if it is a MIN level then you choose all the branches below it and then finally, you return the sub tree that is constructed.

Now, this represents one strategy for Max likewise you can make many choices. So, remember that when you are choosing one branch at any node, then you have a choice point and this is a place where Max has to decide as to which move to make. So, Max will have a set of strategies corresponding to any game tree.

(Refer Slide Time: 17:12)



Two Strategies for Max

Strategies in a game tree are subtrees that represent choices of one player. The figure shows two strategies for Max.

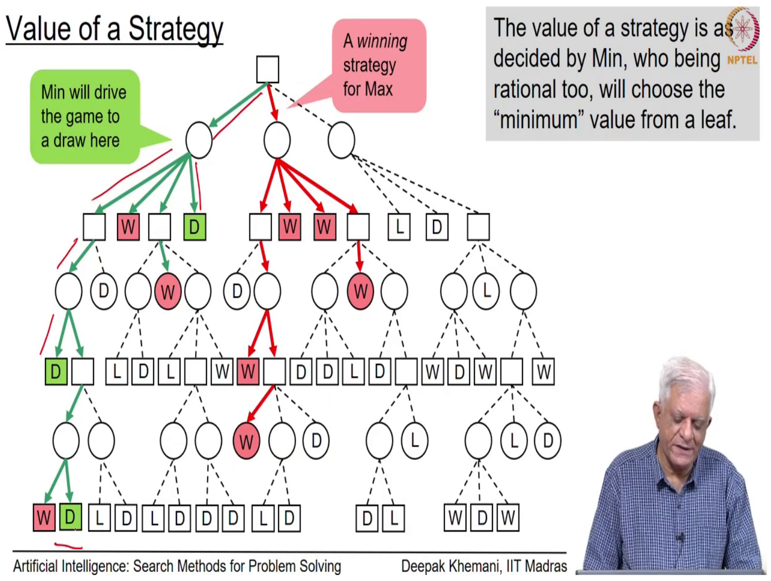
Artificial Intelligence: Search Methods for Problem Solving

Deepak Khemani, IIT Madras

So, here is a example of our game tree. We have two strategies for Max – one is let us call it a red strategy and let us call the other a green strategy. Remember that strategies are sub trees of the game tree. So, what you see is the complete game tree and what you see in red is a sub tree and what you see in green is another sub tree. So, these are two different possible strategies for max.

Obviously, there can be other strategies at as well. At every place that we chose a move for Max for example, this one you could have chose a different move and that would be a different strategy. Now, what do these two these two strategies achieve for max?

(Refer Slide Time: 18:11)


Value of a Strategy
Min will drive the game to a draw here
A winning strategy for Max
The value of a strategy is as decided by Min, who being rational too, will choose the "minimum" value from a leaf.
Artificial Intelligence: Search Methods for Problem Solving
Deepak Khemani, IIT Madras

You can see that, these are the leaves in the two strategies which are coloured and in the case of the green strategy; because now the choices are left to Min; remember that Max has frozen Max's choices and the game will be decided by what Min does. So, clearly if you look at the green sub tree, it has two kinds of nodes either a win for Max or a draw for both of them.

There is no L label leaf here and so, the best that Min can do in this strategy if Max want to choose the green strategy, it would drive the game towards the draw. So, if Max makes this move, then Min can make this move to take the game to a draw or Min can also make this move which will in which Max has says that he will make this move and then Min can again drive it to a draw.

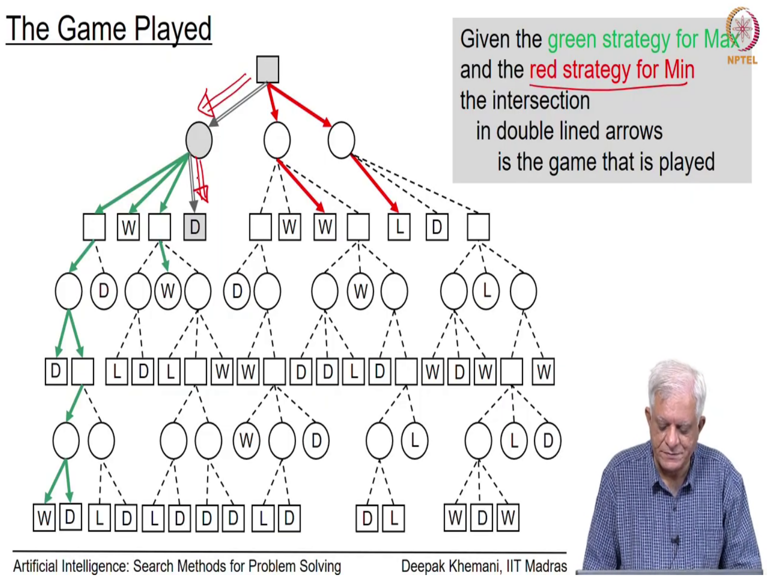
And, likewise for the third node which is at the bottom here essentially. So, clearly Min will not drive the game towards a node label W; because Min is also trying to do the best. In this particular if Max's said that this is my strategy, then Min can say I will accept a draw from you. On the other hand, the red strategy has all the leaves which are labelled with W essentially.

So, Min has no choice, but to lose the game and we saw when we will analyze this particular game tree that the minimax value of this game tree was plus 1. Or it was a win for Max and Max can achieve that by choosing a winning strategy, a winning strategy is a strategy in which all the leaves are labelled with W for max.

So, ideally the task of Max is to analyze the game tree and come up with a winning strategy. And, we will focus a little bit on this specially for game trees which are much larger than this particular small game tree.

Now, if the players have chosen the strategies remember Min can also choose a strategy. Likewise in a strategy for Min you would make one choice for Min node and all choices for Max node. So, Min can also analyze this particular game tree and say that this is my strategy.

(Refer Slide Time: 20:43)



The Game Played

Given the green strategy for Max
and the red strategy for Min
the intersection
in double lined arrows
is the game that is played

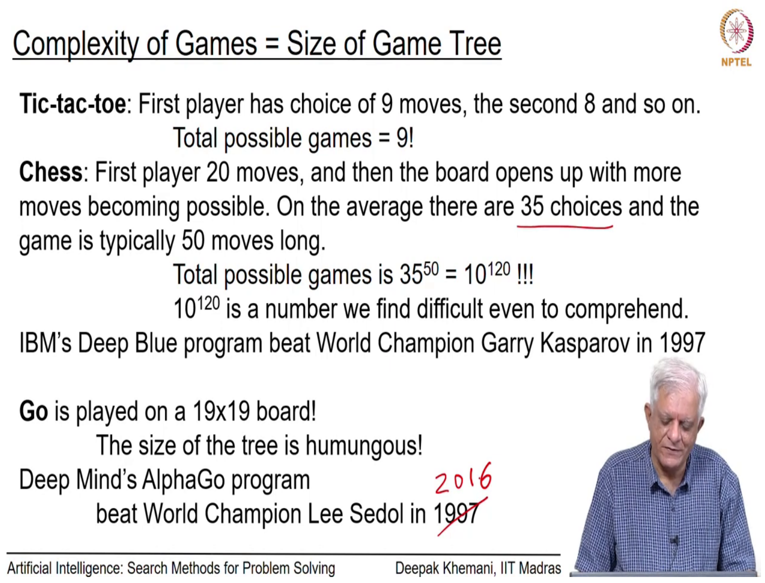
Artificial Intelligence: Search Methods for Problem Solving

Deepak Khemani, IIT Madras

So, let us say now the red sub tree is a strategy for min. So, the red strategy is for Min which means. So, there is a there is a common edge between. So, the green strategy is a strategy for Max, the red strategy is the strategy for Min and the intersection of these two strategies is going to be a path which is depicted by double lines in this tree. And, this path which is the intersection of the two strategies is the game that will actually be played if both the players have frozen their strategies.

So, remember that the red sub tree is the strategy by Min, the green sub tree is a strategy by Max and this paths that we have shown is the intersection of the two strategies. These are the edges which are common to both the strategies. So, once the two players are frozen the strategies, we know what is the game that can be played and what will be the outcome of the game as well.

(Refer Slide Time: 22:01)



Complexity of Games = Size of Game Tree

Tic-tac-toe: First player has choice of 9 moves, the second 8 and so on.
Total possible games = 9!

Chess: First player 20 moves, and then the board opens up with more moves becoming possible. On the average there are 35 choices and the game is typically 50 moves long.
Total possible games is $35^{50} = 10^{120}$!!!
10^{120} is a number we find difficult even to comprehend.
IBM's Deep Blue program beat World Champion Garry Kasparov in 1997

Go is played on a 19x19 board!
The size of the tree is humungous!
Deep Mind's AlphaGo program
beat World Champion Lee Sedol in ~~1997~~ 2016

Artificial Intelligence: Search Methods for Problem Solving Deepak Khemani, IIT Madras

Now let us have a brief look at complexity of games and when we say complexity of games. We mean the size of the game tree. Of course, this is complexity not in terms of the order notation that we use in computer science very often where the complexity is in terms of some parameter like the size of a matrix.

For example, but here the size of the game is fixed in the sense that the board is fixed, we simply want to have a estimate of what is the size of the game tree and this is what we mean by complexity of a game tree.

So, the simplest possible game is tic-tac-toe or cross and noughts that we have seen a little while before you know that the first player has 9 moves. You can place a cross in any of the 9 squares and then the second player would have 8 moves and so on.

So, we are not kind of taking care of symmetries which may or may not be easy to handle in a program. But in the worst case you can say that, the first player has 9 choices in practice if you take into account symmetries the first player will have only 3 choices either a corner or the center or the side.

But, then we want to simplify matters. So, we say the first player has 9 choices second player has 8 and so on and the total possible games will be 9 factorial; which as we have discussed we already know that always ends in a draw when both the players are playing perfectly.

Let us look at the game of chess. Now, the branching factor of chess is also variable. So, if you know the first player white in the case of chess can make 20 moves. It can make 8 pawn moves moving them forward one step another 8 pawn moves moving them forward 2 steps then 16. And it can move each of the knight forward and left and forward and right. So, that is 4 moves. So, the total moves are 20.

Any program which searches the entire tree will have to consider all possible 20 moves. So, anyone plays do not tend to do that you know, we always go through a process of learning in which we figured out that certain openings are good and in fact, there are books available on chess like modern chess openings which tell you what are the different kinds of openings that have been used by good players.

But, as the game progresses the board opens up because as pawns move forward the species behind them. For example, the queen or the rook or the bishop they become more mobile and the number of moves becomes larger and larger and it increases towards what is called as a middle game. Then towards the end game when the number of species start declining because many captures are taking place the number of lancing factor again goes down a little bit.

It has been estimated that on the average the lansing factor in the game of chess is 35. So, on the average there are 35 choices that a player can make and typically a game is 50 moves long essentially. So, if we take these average numbers, then you can imagine that there are 35 ways to 50 possible games and which translates to 10 raise to 120 possible games.

Now, this 10 raise to 120 is a number that we cannot even begin to comprehend essentially. If you just think of the fact that the number of fundamental particles in the entire universe is estimated to be 10 raised to 70 or 10 raised to 80 or something.

And each if each of these fundamental particles was a supercomputer looking at whatever you want 10 billion moves a second, 10 trillion moves a second you can see that there would still be a factor of 10 raised to 30 – 40 remaining and that would translates into millions and millions and millions of centuries.

So, clearly we cannot analyze the full chess tree. And therefore, we do not know what is the value of a chess game. Which means that we do not know whether if both the players are playing perfectly; whether white will always win or whether the game will always be a draw or whether black will always win and that is why chess is still a fascinating game for us unlike Tic-tac-toe or cross and noughts which we know will always be a draw.

Now, we will know that chess has been it is not been solved in the sense we do not know the value of the game. But we have programs which are good enough to play to be the best humans and that started off in 1997 when IBM's Deep Blue program beat the world champion Garry Kasparov essentially.

(Refer Slide Time: 27:49)

AlphaGo beats World Champion at Go in 2016

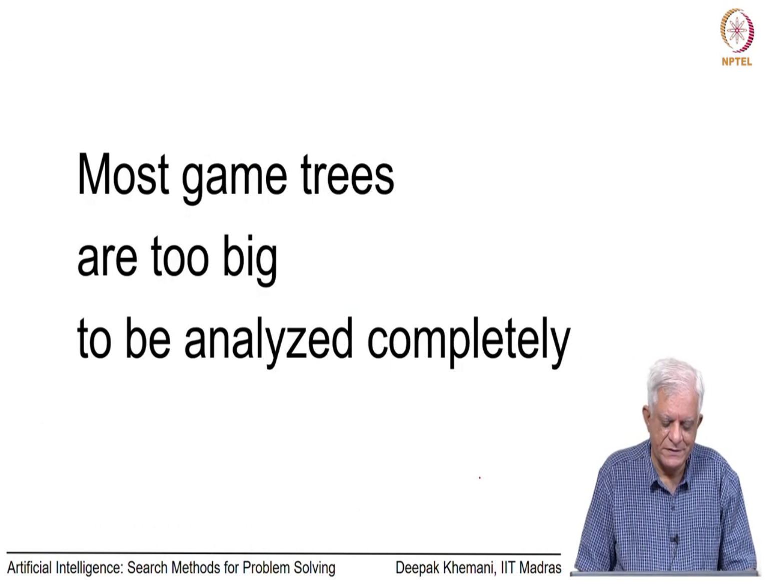
from this Popular Mechanics article

Now, the next board game which for many years was considered to be difficult for machines to tackle was go because Go was played on a 19 by 19 chess board. So, in some sense Go and Tic-tac-toe have a similarity. In that end Go if you remember the figure that we had seen you had to place either a black coin or a white coin on the board.

And, there were 19 by 19 choices in the first place and you can imagine that the size of the branching tree was really humungous. And, for a long time people thought that Go is difficult for a computers to beat humans act and people said that Go involves a different kind of reasoning which some people associated with Zen reasoning where you know you kind of look at the chess board or a go board for a long time and somehow the right move pops up.

But, as we as history has shown us the program Alphago written by deep mind it did beat the world champion in this is a mistake in 2016.

(Refer Slide Time: 29:13)



Most game trees
are too big
to be analyzed completely

Artificial Intelligence: Search Methods for Problem Solving Deepak Khemani, IIT Madras

So, Go is also a board game it has been tackled by machines essentially. How do this programs work? As you have just seen most interesting games they are too huge for to be analyzed completely. The chess she has about 10 raised to 120 possible games the go game has much more and therefore, you cannot analyze the game completely.

So, now, we will start focusing on how to write programs to play these games and as we will see these programs will rely on a limited look ahead that you do not look ahead until the end of the game, but only for a few moves ahead. And, if you do and when you do that you apply what we will call as a an evaluation function which is going to be a static function which will look at a board position and give it a value essentially.

So, this will be a value which would signify whether the game is good for player Max or whether it is good for player Min and so on. And, we would use this evaluation function to guide ours game playing program and we will do that in the next session onwards and we will look at a few algorithms for playing real games.

So, see you the next time.